

- 2 -

IN THE CLAIMS:

Amended claims follow:

1. (Currently Amended) A method for accelerated scanning, comprising:
 - (a) identifying a file access pattern associated with data;
 - (b) reading the data based on the file access pattern; and
 - (c) performing a virus scan on the data;wherein the data is read and scanned by executing a first thread of operation for reading the data and a second thread of operation for scanning the data;
wherein the first thread of operation is executed in parallel with the second thread such that, while a first portion of the data is being scanned, a second portion of the data to be scanned is being read and cached, so that, when a scanner is available for scanning the second portion of the data, a delay associated with reading the second portion of the data is avoided;
wherein if it is determined that the data does not have the associated file access pattern, the data is read and the virus scan is performed, after which it is determined whether the virus scan was slower than a predetermined amount;
wherein the file access pattern is conditionally generated based on whether the virus scan was slower than the predetermined amount;
wherein the file access pattern is not generated if the virus scan was slower than the predetermined amount in order to save space in a file access pattern database.
2. (Original) The method as recited in claim 1, wherein if it is determined that the data does not have the associated file access pattern, the data is read and a file access pattern associated with the file is generated and stored.
3. (Cancelled)
4. (Cancelled)

- 3 -

5. (Original) The method as recited in claim 1, wherein the file access pattern includes a file location.
6. (Original) The method as recited in claim 1, wherein the file access pattern includes a data amount.
7. (Cancelled)
8. (Previously Presented) The method as recited in claim 1, wherein the first thread of operation includes retrieving the file access pattern, reading the data based on the file access pattern, and caching the data.
9. (Original) The method as recited in claim 8, wherein the second thread of operation includes determining whether the file access pattern is valid, and reading the data from the cache if it is determined that the file access pattern is valid.
10. (Original) The method as recited in claim 8, wherein the second thread of operation includes determining whether the data is available for being read from the cache, and reading the data if the data is available for being read from the cache.
11. (Original) The method as recited in claim 1, and further comprising determining whether the file access pattern is invalid.
12. (Original) The method as recited in claim 11, and further comprising deleting the file access pattern if the file access pattern is determined to be invalid.
13. (Original) The method as recited in claim 12, and further comprising reading the data and generating a file access pattern associated with the file if the file access pattern is deleted.

- 4 -

14. (Currently Amended) A computer program product for accelerated scanning, comprising:

- (a) computer code for identifying a file access pattern associated with data;
- (b) computer code for reading the data based on the file access pattern; and
- (c) computer code for performing a virus scan on the data;

wherein the data is read and scanned by executing a first thread of operation for reading the data and a second thread of operation for scanning the data;

wherein the first thread of operation is executed in parallel with the second thread such that, while a first portion of the data is being scanned, a second portion of the data to be scanned is being read and cached, so that, when a scanner is available for scanning the second portion of the data, a delay associated with reading the second portion of the data is avoided;

wherein if it is determined that the data does not have the associated file access pattern, the data is read and the virus scan is performed, and it is determined whether the virus scan is slower than a predetermined amount;

wherein the file access pattern is conditionally generated based on whether the virus scan is slower than the predetermined amount;

wherein the file access pattern is not generated if the virus scan is slower than the predetermined amount in order to save space in a file access pattern database.

15. (Original) The computer program product as recited in claim 14, wherein if it is determined that the data does not have the associated file access pattern, the data is read and a file access pattern associated with the file is generated and stored.

16. (Cancelled)

17. (Cancelled)

18. (Original) The computer program product as recited in claim 14, wherein the file access pattern includes a file location.

- 5 -

19. (Original) The computer program product as recited in claim 14, wherein the file access pattern includes a data amount.
20. (Cancelled)
21. (Previously Presented) The computer program product as recited in claim 14, wherein the first thread of operation includes retrieving the file access pattern, reading the data based on the file access pattern, and caching the data.
22. (Original) The computer program product as recited in claim 21, wherein the second thread of operation includes determining whether the file access pattern is valid, and reading the data from the cache if it is determined that the file access pattern is valid.
23. (Original) The computer program product as recited in claim 21, wherein the second thread of operation includes determining whether the data is available for being read from the cache, and reading the data if the data is available for being read from the cache.
24. (Original) The computer program product as recited in claim 14, and further comprising computer code for determining whether the file access pattern is invalid.
25. (Original) The computer program product as recited in claim 24, and further comprising computer code for deleting the file access pattern if the file access pattern is determined to be invalid.
26. (Original) The computer program product as recited in claim 25, and further comprising computer code for reading the data and generating a file access pattern associated with the file if the file access pattern is deleted.
27. (Currently Amended) A system for accelerated scanning, comprising:
- (a) logic for identifying a file access pattern associated with data;

- 6 -

- (b) logic for reading the data based on the file access pattern; and
- (c) logic for performing a scan on the data;

wherein the data is read and scanned by executing a first thread of operation for reading the data and a second thread of operation for scanning the data;

wherein the first thread of operation is executed in parallel with the second thread such that, while a first portion of the data is being scanned, a second portion of the data to be scanned is being read and cached, so that, when a scanner is available for scanning the second portion of the data, a delay associated with reading the second portion of the data is avoided;

wherein if it is determined that the data does not have the associated file access pattern, the data is read and the scan is performed, after which it is determined whether the scan was slower than a predetermined amount;

wherein the file access pattern is conditionally generated based on whether the scan was slower than the predetermined amount;

wherein the file access pattern is not generated if the scan was slower than the predetermined amount in order to save space in a file access pattern database.

28. (Cancelled)

29. (Cancelled)

30. (Currently Amended) A method for reducing delay associated with reading data from memory during a scan, comprising:

- (a) initiating a scan;
- (b) identifying a file to be scanned during the scan;
- (c) determining whether the file has a file access pattern associated therewith;
- (d) if it is determined that the file does not have the associated file access pattern,
 - (i) reading the data from the file,
 - (ii) scanning the data,
 - (iii) identifying a file access pattern associated with the file, and
 - (iv) storing the file access pattern;

- 7 -

(e) if it is determined that the file has the associated file access pattern, executing a first thread of operation and a second thread of operation, the first thread of operation including:

- (i) retrieving the file access pattern,
- (ii) identifying a file location and data amount of the file access pattern,
- (iii) reading data from the file associated with the identified file location and data amount,

- (iv) caching the data, and
- (iv) repeating (ii)-(iv); and

(f) said second thread of operation including:

- (i) determining whether the file location and data amount are valid,
- (ii) deleting at least a portion of the file access pattern associated with the file if it is determined that the file location and data amount are not valid,
- (iii) determining whether the data is available for being read from the cache,
- (iv) reading the data from the cache if it is determined that the file location and data amount are valid, and the data is available for being read from the cache, and
- (v) scanning the data;

wherein the first thread of operation is executed in parallel with the second thread such that, while a first portion of the data is being scanned, a second portion of the data to be scanned is being read and cached, so that, when a scanner is available for scanning the second portion of the data, a delay associated with reading the second portion of the data is avoided;

wherein if it is determined that the data does not have the associated file access pattern, the data is read and the scan is performed, after which it is determined whether the scan was slower than a predetermined amount;

wherein the file access pattern is conditionally generated based on whether the scan was slower than the predetermined amount;

wherein the file access pattern is not generated if the scan was slower than the predetermined amount in order to save space in a file access pattern database.

- 8 -

31. (Currently Amended) The method as recited in claim 1, wherein the file access pattern is stored in a database including a plurality of file access patterns, the file access pattern database, each of the plurality of file access patterns located in the file access pattern database being associated with a different set of data.

32. (Previously Presented) The method as recited in claim 1, wherein the file access pattern includes a checksum of a file path and a file name.

33. (Previously Presented) The method as recited in claim 1, wherein the data is stored in a cache and the cache is continuously monitored for determining if data is available for scanning.

34. (Cancelled)

35. (Previously Presented) The method as recited in claim 1, wherein the file access pattern is stored in a binary format.